1 Listing of Algorithms

1.1 *Computation of Busy-Hour Path Delay for Samples*

Algorithm No. 1

Computation of Average Path Delay for Samples and Detection of Delay Busy-Hour Period ($1 \leq \Psi \leq 24$ Hours)

****

Start

For i = 1 To m — *scan paths*

- $A_i = 0$ — *initialize Sum of Delays in Window*
- $B_i = 0$ — *initialize Maximum Sum of Delays*
- $D_i = 0$ — *initialize Average Delay*
- $k_{max} = 0$ — *initialize first ordinal number of busiest-hour(s)-period timestamp*

For k = 1 To $[(24 * 3600)/\delta] - (\omega * \Psi)$ — *ordinal number of timestamp*

For j = 1 To n — *hop number*

For h = k To $[k - 1 + (\omega * \Psi)]$

$A_i = A_{ijk} + A_i$ — *add samples in $\omega * \Psi$ window.*

Next h

Next j

If $B_i < A_i$ **Then** — *Select higher of two values.*

$B_i = A_i$

$K_{max} = k$ — *Capture the first ordinal number of busiest-hour(s)-period timestamp.*

End If

Next k

$D_i = \lceil \{[B_i/(\omega * \Psi * j)]\} \rceil$ — *Calculate average: divide sum by*

number of samples and hops.
Compute ceiling.

Next i
End
****

1.2 *Data Sieving*

Algorithm No. 2

Data Sieving for Path Delay of Samples

****

1. Remove delay values for paths containing two nodes AND these two nodes are located in the same site (e.g., node LAX1 and node LAX2). Delay in the direction from Node11 to Node1p is designated Bi+ and delay in the direction from Node1p to Node11 is designated Bi–.

(Delays of trunks within a hub shall be eliminated, e.g., LAX1-LAX2).

1.

2. Remove delay values for B_i^+ if $B_i^+ \geq B_i^-$.

(For each edge node pair, the algorithm shall eliminate one of two paths with the higher delay and also eliminates the edge node pairs with equal delay. Edge node pairs with higher delay are eliminated to exclude paths that were formed due to a network failure and those with asymmetric routes.).

3. Remove delay values for B_i^- if $B_i^- \geq B_i^+$.
****
******

1.3 *Standard Deviation - Root Mean Square (RMS)*

Definitions
S_{ij} = Standard Deviation of Path i, Hop j
SQRT = Square Root

Algorithm No. 3

Computation of the Standard Deviation for Trunks within Paths

****

Start

$D_{ij} = 0$ — *Initialize average busy-hour delay. for trunk*

$S_{ij} = 0$ — *Initialize*

For i = 1 To m — *scan paths*

  For j = 1 To n — *hop number*

    For k = K_i To $(K_i + \omega * \Psi)$ — *Read K_i from output of Algorithm 2.*

      $D_{ij} = (D_{ij} + A_{ij})/(\omega * \Psi)$ — *Compute average value for busy hour.*

    Next k

    For k = K_i To $[K_i + (\omega * \Psi)]$ — *Read K_i from output of Algorithm 2.*

      $S_{ij} = S_{ij} + (D_{ij} - A_{ij})^2$

    Next k

$$S_{ij} = \text{SQRT}\left[\frac{S_{ij}}{(\omega * \Psi) - 1}\right]$$

  Next j

Next i

End

****

1.4 *Computation of the Standard Deviation for a Path*

****

$$\sigma_{path} = \{\sigma_{Trunk1}{}^2 + \sigma_{Trunk\,2}{}^2 + \ldots + \sigma_{Trunk\,N}{}^2)\}^{1/2}$$

****

1.5 *Computation of Busy-Hour Trunk Delay for Samples*

An explanation of the this scheme as well as a description of the relationships of data elements can be best described by a two-dimensional matrix of the order m * p that includes scalar components. There are two 2D matrices per traffic class

$$\begin{matrix} A_{11} & A_{12} & \ldots & A_{1p} \\ A_{21} & A_{22} & \ldots & A_{2p} \\ \ldots & \ldots & \ldots & \ldots \\ A_{m1} & A_{m2} & \ldots & A_{mp} \end{matrix}$$

2D Delay Matrix

Order of matrix = $m * p \approx 5.4 * 10^3$

Definition of Matrix Elements

A_{ik}	Sample of delay value
Units	[milliseconds]
Quantity	Scalar
A_{ik}	[Delay Sample $\mid_i$]

i	Trunk Number	
Units	[Dimensionless]	
Domain	$1 \leq i \leq m$	
$m \approx$	187	(e.g., assume 153 trunks per priority traffic class + 34 intra-site trunks)

k	Ordinal number of timestamp
Units	[Dimensionless]
Domain	$1 \leq k \leq (24 * 3600)/ \delta$
$p \approx$	288 ((provided for reference only, e.g., 12 samples per hour x 24 hours per day = 288 samples per 24 hours)

δ	Sampling interval
Units	[seconds]
Domain	300 seconds $\leq \delta \leq$ 600 seconds

ω	Number of samples per Hour
$\omega =$	$f(\delta)$

Ψ	Period of averaging delay [Hour]
Domain	$1 \leq \Psi \leq 24$
Average	Arithmetic average

A_i = Intermediate aggregated delay
B_i = Intermediate aggregated delay
D_i = Average delay for trunk I

K_{max} = First ordinal number of timestamp of busiest hour(s) period in a 24-hour period. This value is required for computing the Standard Deviation for trunks and the 95 percentile delay.

Algorithm No. 5

Computation of Average Trunk Delay for *Samples* and Detection of Delay Busy-Hour ($1 \leq \Psi \leq 24$ Hours)

```
*****************************************************************************
Start

For i = 1 To m

    A_i = 0          initialize
    B_i = 0          initialize
    D_i = 0          initialize
    k_max = 0        initialize first ordinal number for the nine-busiest-consecutive -
                     hour- period timestamp

    For k = 1 To [(24 * 3600)/ δ] – (ω * Ψ)        ordinal number of timestamp

        For h = k To [k - 1 + (ω * Ψ)]

            A_i = A_ik + A_i                       add samples in ω*Ψ window.

        Next h

            If B_i < A_i Then

                B_i = A_i
                K_max = k                          Capture the first ordinal number of
                                                   busiest-hour(s)-period timestamp

            End If

    Next k

    D_i = ⌈[B_i/(ω * Ψ)]⌉                          Calculate average: Divide sum by
                                                   number of samples and compute
                                                   ceiling.

Next i

End
*****************************************************************************
```

1.6 *Standard Deviation for Trunks*

Algorithm No. 6

Computation of the Standard Deviation for Trunks Only

****

Start

$D_i = 0$ *Initialize Average Delay for Busy Hour*

$S_i = 0$ *Initialize Standard Deviation for Trunk Delay*

For $i = 1$ To m *scan trunks*

For $k = K_i$ To $(K_i + \omega * \Psi)$ *Compute average value for busy hour*

$$D_i = (D_i + A_i)/(\omega * \Psi)$$

Next k

For $k = K_i$ To $(K_i + \omega\Psi)$ *first ordinal number of time stamp for busy hour*

$$S_i = S_i + (D_i - A_i)^2$$

Next k

$$S_i = \mathrm{SQRT}\left[\frac{S_i}{(\omega * \Psi) - 1}\right]$$

Next i

End

****

1.7 *Computation of Upper Bound of Confidence Interval*

Algorithm No. 7

Computation of Upper Bound of Confidence Interval

****

δ = Sampling interval
σ_{path} = Standard Deviation of Path Delay

Start

Upper Bound of Confidence Interval = $Z * \dfrac{\sigma_{path}}{(3600/\delta)^{1/2}}$

End
****

1 - α Confidence Level	Z
90%	1.645
91%	1.695
92%	1.751
93%	1.812
94%	1.881
95%	1.960
96%	2.054
97%	2.170
98%	2.326
99%	2.576
99.9%	3.291
99.99%	3.891

Confidence Level versus variable Z

1.8 *Algorithm for Calculating Busy-Hour Path Delay for the Population*

Algorithm No. 8

Computation of Busy-Hour Path Delay for the *Population* of Traffic

****

Start

1. Calculate busy-hour path delay for samples. Reference: Algorithm 2

2. Calculate busy-hour path delay for the population:

> **Busy-Hour Delay for Path Samples + Upper Bound of Confidence Interval**

End

****

1.9 *Busy-Hour Trunk Delay for Traffic Population*

Algorithm No. 9

Computation of Busy-Hour Trunk Delay for the *Population* of Traffic

****

Start

1. Calculate busy-hour trunk delay for samples. Reference: Algorithm 5

2. Calculate busy-hour trunk delay for the population:

> **Busy-Hour Trunk Delay for Samples + Upper Bound of Confidence Interval**

End

****

Algorithm No. 9 will be used for producing the delay baseline for trunks.

1.10 *Algorithm for Calculating Percentile Delay*

Example: Calculate the 95^{th} percentile delay

Several percentile delay lookup tables are provided for ease of calculation and for reducing calculation time: 92, 95, and 98 percentile delay. Other percentile table can be provided for any desired resolution (maximum of 14 significant figures).

The following table is presented here for demonstrating the method of calculation

CoV --->	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
Ratio --->	1.015	1.03	1.045	1.065	1.08	1.1	1.115	1.135	1.15
CoV --->	0.11	0.12	0.13	0.14	0.15	0.16	0.17	0.18	0.19
Ratio --->	1 185	1.205	1.22	1.24	1.255	1.275	1.295	1.31	1.33
CoV --->	0.21	0.22	0.23	0.24	0.25	0.26	0.27	0.28	0.29
Ratio --->	1.365	1.385	1 405	1.42	1.44	1.46	1.48	1 5	1.515
CoV --->	0.31	0.32	0.33	0.34	0.35	0.36	0.37	0.38	0.39
Ratio --->	1.555	1.575	1.595	1.615	1.635	1.655	1.675	1.695	1.715

Part of 95 Percentile Lookup Table

Algorithm No. 10

Computation of Percentile Delay

****

A. Calculate average delay for samples for the nine busiest consecutive hours in a 24-hour period. Reference: Algorithm 2.

B. Trunks

1. Standard Deviation of Delay Samples for Trunks

Calculate the Standard Deviation for the nine busiest consecutive hours in a 24-hour period. Reference: algorithm 6.

2. $$CoV_{Trunk9B} = \frac{\sigma_{Trunk}}{\text{average delay for samples of nine busiest consecutive hours in a 24-hour period}}$$

3. $$CoV_{Trunk1B} = \frac{\sigma_{Trunk}}{\text{average delay for samples of busiest hour in a 24-hour period}}$$

Note: Items B. 2. is computed for direct output report and as an intermediate step for subsequent computation. Item B. 3. is computed for a direct output report, not as an intermediate step for other computations. Reference: algorithm 5.

Reference: Algorithm No. 5

C. Paths

1. Standard Deviation of Paths

> **Calculate the Standard Deviation for the nine busiest consecutive hours in a 24-hour period. Reference: algorithm 4.**

2. $$CoV_{Path} = \frac{\sigma_{path}}{\text{average delay for samples of nine busiest consecutive hours in a 24-hour period}}$$

Reference: Algorithm No. 2
Notes: 1B = Busiest Hour; 9B = Nine Busiest Hours

D. Look up Ratio in table, for CoV of steps B.2. and C.2. above:

1. Trunks

> **95 Percentile Delay of Trunks =**
>
> > **⌈[average trunk delay for samples of nine busiest consecutive hours in a 24-hour period * Ratio]⌉**

2. Paths

> **95^{th} Percentile Delay of Trunks =**
>
> > **⌈[average path delay for samples of nine busiest consecutive hours in a 24-hour period * Ratio]⌉**

End
****

1.11 *Algorithm for Establishing Delay Baseline for Network Management*

Algorithm No. 13

****
Start

1. Collect daily average values of busy-hour delay for the population for one calendar month

2. Compute the 95 percentile of daily delay values

3. Perform calculation monthly.

End
****

After this algorithm is implemented, tested, and validated, a similar or modified method can be applied to the delay baseline for SLAs.

1.12 *Algorithm for Computing Exception of Delay Values*

Exceptions: Daily Delay vs. Baseline Delay for Network Management for Trunks and Paths)

Algorithm No. 14

Exceptions: Daily Delay vs. Baseline Delay for Network Management for Trunks and Paths

****
Start

1. For each city pair, compare daily delay value with delay value of delay baseline

2. $$\text{Excess of Delay Ratio} = \left\lceil \left[\frac{\text{City-Pair daily delay} - \text{baseline delay of previous month}}{\text{Baseline delay}} * 100\% \right] \right\rceil$$

3. If Excess of Delay Ratio ≥ ExceptionThreshold Then

    List traffic class, city pair, and Excess of Delay Ratio

End
****

Variable: **ExceptionThreshold** = 20%

Reference: Algorithms No. 2, 5, and 13

1.13 *Algorithm for Computing Risk Value*

Algorithm No. 15

Risk value = (1 – Confidence Coefficient)/2

This is the probability of exceeding estimated delay of the traffic population.

1.14 *Exceptions: Coefficient of Variation of Delay for Trunks*

Algorithm No. 16

****
Start

1. Compute the Coefficient of Variation of delay for each trunk (Ref: Algorithm No. 10, part 2)

2.

Condition	**Result**
If $0.5 \leq \text{CoV} < 1.0$	Then CoV Alert Stage = Minor
If $1.0 \leq \text{CoV} < 2.0$	Then CoV Alert Stage = Major
If $\text{CoV} \geq 2.0$	Then CoV Alert Stage = Critical

Table 1: CoV alert conditions

End
****

1.15 *Fixed Delay*

Fixed delay is the minimum possible delay along a network path, when queuing delay is not present (buffers of ATM switches are empty).

The fixed delay can be computed by extrapolating delay from delay measurements on two sequential days.

$$\text{Fixed Delay} = \tfrac{1}{2}\left[D_1 + D_2 - T_S\left(\frac{U_1}{1 - U_1} + \frac{U_2}{1 - U_2}\right)\right]$$

where

D_1 =	Average delay for period of time T on day 1
D_2 =	Average delay for period of time T on day 2
U_1 =	Average link utilization for period of time T on day 1
U_2 =	Average link utilization for period of time T on day 2
T_S =	Serialization delay for link

Measurements can be taken on consecutive dates for a period of T = 3 hours.

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

Application deficiencies found during scanning:

☐ Page(s)__________ of ______________________________ were not present for scanning.
(Document title)

☐ Page(s)__________ of ______________________________ were not present for scanning.
(Document title)

☒ ***Scanned copy is best available.*** Drawing figures 1 & 5 are dark.